Do it Once. Do it Right.
MAGIC SOFTWARE

uniPaaS Partitioning Troubleshooting Guide

uniPaaS
The Power of Choice
Enabled by eDeveloper

Magic Software Enterprises Ltd., 5 Haplada Street, Or-Yehuda 60218, Israel
Tel: +972 (0)3 538 9292 | Fax: +972 (0)3 538 9333, +972 (0)3 538 9393 | info@magicsoftware.com | www.magicsoftware.com

The information in this manual/document is subject to change without prior notice and does not represent a commitmehnt on the part of Magic Software Enterprises Ltd.

Magic Software Enterprises Ltd. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose.

The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms and conditions of the license agreement. It is against the law to copy the software on any medium except as specifically allowed in the license agreement.

No part of this manual and/or databases may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or information recording and retrieval systems, for any purpose other than the purchaser's personal use, without the prior express written permission of Magic Software Enterprises Ltd.

All references made to third-party trademarks are for informational purposes only regarding compatibility with the products of Magic Software Enterprises Ltd.

Unless otherwise noted, all names of companies, products, street addresses, and persons contained herein are part of a completely fictitious scenario or scenarios and are designed solely to document the use of uniPaaS.

Magic® is a registered trademark of Magic Software Enterprises Ltd.
Btrieve® and Pervasive.SQL® are registered trademarks of Pervasive Software, Inc.
IBM®, Topview™, iSeries™, System i™, pSeries®, xSeries®, RISC System/6000®, DB2®, and WebSphere® are trademarks or registered trademarks of IBM Corporation.
Microsoft®, FrontPage®, Windows™, WindowsNT™, and ActiveX™ are trademarks or registered trademarks of Microsoft Corporation.
Oracle® and OC4J® are registered trademarks of the Oracle Corporation and/or its affiliates.
Linux® is a registered trademark of Linus Torvalds.
UNIX® is a registered trademark of UNIX System Laboratories.
GLOBEtrotter® and FLEXlm® are registered trademarks of Macrovision Corporation.
Solaris™ and Sun ONE™ are trademarks of Sun Microsystems, Inc.
HP-UX® is a registered trademark of the Hewlett-Packard Company.
Red Hat® is a registered trademark of Red Hat, Inc.
WebLogic® is a registered trademark of BEA Systems.
Interstage® is a registered trademark of the Fujitsu Software Corporation.
JBoss™ is a trademark of JBoss Inc.
Systinet™ is a trademark of Systinet Corporation.
Portions Copyright © 2002 James W. Newkirk, Michael C. Two, Alexei A. Vorontsov or Copyright © 2000-2002 Philip A. Craig

Clip art images copyright by Presentation Task Force®, a registered trademark of New Vision Technologies Inc.
This product uses the FreeImage open source image library. See http://freeimage.sourceforge.net for details.
This product includes software developed by the Apache Software Foundation (http://www.apache.org/).
This product includes software developed by Computing Services at Carnegie Mellon University (http://www.cmu.edu/computing/). Copyright © 1989, 1991, 1992, 2001 Carnegie Mellon University. All rights reserved.
This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).
This product includes software that is Copyright © 1998, 1999, 2000 of the Thai Open Source Software Center Ltd. and Clark Cooper.
This product includes software that is Copyright © 2001-2002 of Networks Associates Technology, Inc All rights reserved.
This product includes software that is Copyright © 2001-2002 of Cambridge Broadband Ltd. All rights reserved.
This product includes software that is Copyright © 1999-2001 of The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved.

All other product names are trademarks or registered trademarks of their respective holders.

uniPaaS Troubleshooting Guide
July 2009

Copyright © 2009 by Magic Software Enterprises Ltd. All rights reserved.

uniPaaS Partitioning Troubleshooting

1 OVERVIEW 4

2 TERMINOLOGY 4

3 UNIPAAS PARTITIONING MODULES 4

3.1 PORTS 5

3.2 RESOLVING HOST NAMES 5

3.3 HANDLING CONNECTIONS 5

3.4 TIMEOUTS 6

3.5 LOG FILE SETTINGS 6

APPENDIX I - UNIPAAS'S INFORMATION AND ERROR CODES 7

INFORMATION 7

ERRORS 9

APPENDIX II - DB ERRORS 16

APPENDIX III - WINSOCK ERRORS 18

APPENDIX IV - TEST CASES 21

-105: BROKER NOT RESPONDING 21

PROCEDURE 21

-138: RUNTIME CRASH 21

PROCEDURE 21

-144: LOW-LEVEL CONNECTION RESET 22

PROCEDURE 22

-197: CONTEXT NOT FOUND 22

PROCEDURE 22

1 Overview

This document will help troubleshoot certain situations and provide a better understanding of how each of the uniPaaS components interacts with one another.

The topics that will be discussed include:

- How the uniPaaS partitioning modules interact
- Ports
- Resolving host names
- Handling connections
- Timeouts
- Log file settings

The document will also explain the meaning of some of the requester's error codes and will offer guidelines for solving certain situations.

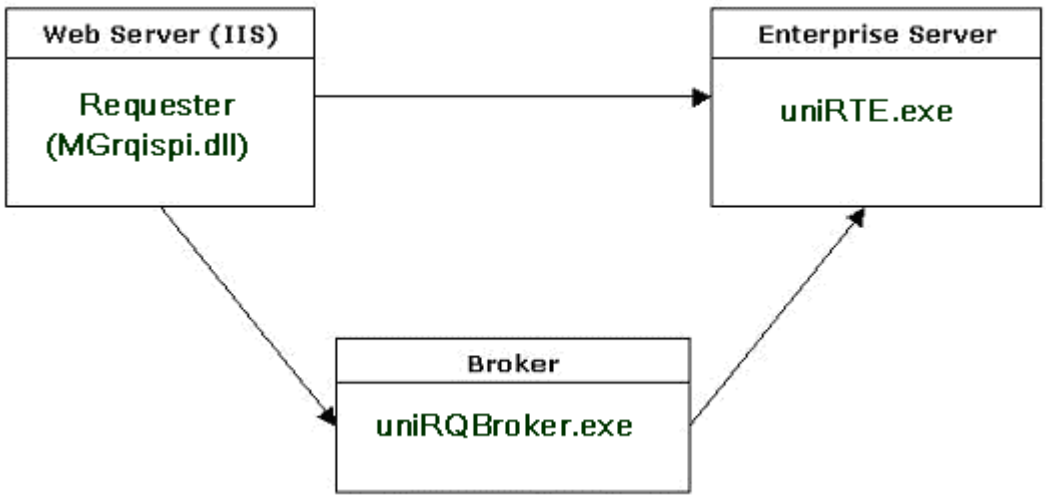
2 Terminology

The TCP/IP Stack refers to the TCP/IP software at the operating system level, and has its own settings, registry, and configuration files. On Windows platforms, the TCP/IP Stack is usually known as Winsock. The TCP/IP stack has several vendors in addition to Microsoft. Errors returned from the TCP/IP Stack are mapped to the uniPaaS partitioning errors, listed in Appendix I - uniPaaS's Information and Error Codes.

3 uniPaaS Partitioning Modules

A typical configuration consists of one broker, one or more enterprise servers, and an ISAPI requester.

This can be illustrated graphically as shown below:



When a request is made to the Web server, the requester polls the broker to find a uniPaaS engine to work with. The broker finds an engine that is not busy, and informs the requester as to which engine is available. If the request is synchronous, the requester then works directly with that engine, and the broker is no longer involved. Otherwise, the broker works directly with the engine, and the requester is no longer involved.

The INI files of each of the components define the ports where the components listen to one another.

3.1 Ports

The broker uses **one** port, the port defined in the BrokerPort section of the Mgrb.ini file.

An enterprise server also uses only **one** port for standard broker-related requests.

However, an enterprise server may use another port for J2EE requests, where the EJBs interact directly with the enterprise server. This is the case when the **Mgreq.ini** file includes the lines below:

```
[MAGIC_MESSAGING_GATEWAYS]

MGSRVR05 = , , , ,MaxThreads=
```

In that case, the enterprise server tries to use the first port defined in the range of ports in the TCP/IP section of the Magic.ini file ([MAGIC_COMMS] TCP/IP - the default is 1500-2000). This eases the integration with EJBs, for which the default port is also 1500.

3.2 Resolving Host Names

This section highlights a number of points about the way TCP/IP host names can be resolved.

The uniPaaS partitioning architecture lets the different modules (broker, clients, and enterprise servers) reside on different computers.

For that purpose, each computer must know the names of the other computers with which it interacts. For example, let's say the broker is positioned on one computer, named SRVR_1, and there are two enterprise servers on two different computers, SRVR_2 and SRVR_3.

The enterprise servers identify themselves to the brokers using the names SRVR_2 and SRVR_3, and these are the names that the broker passes on to clients when the clients send synchronous requests. Therefore, each client must know how to resolve the names SRVR_1, SRVR_2, and SRVR_3.

The best way to accomplish this is to use DNS (Domain Names Service) or DHCP, a centralized pool of names and their known IP addresses. Another way, old-fashioned but simple, is to use a hosts file. The hosts file mechanism requires each computer to have an up-to-date copy of the same hosts file.

3.3 Handling Connections

In the diagram above, arrows represent connections that remain as long as both sides of the connection are alive and functioning. The operating system "netstat" command helps view these connections during different phases of the TCP/IP state diagram.

A requester, such as ISAPI, initially starts a connection to the broker and to each enterprise server. These connections stay ESTABLISHED until the requester or the partner, that is the broker or an enterprise server, is shut down.

When a requester needs to send a request to the broker or an enterprise server and all established connections are already in use, the requester opens a new connection and keeps it ESTABLISHED, as described above. This means that the number of established connections grows gradually until it reaches a maximum number of connections, and then the existing connections serve all further requests without opening new connections. There is only one connection from each enterprise server to the broker.

If the requester shuts down, such as when an IIS restarts, the requester closes its connections to both the broker and all connected enterprise servers gracefully. Then the requester starts opening new connections, exactly as described previously.

If the broker or enterprise servers are shut down without the requester knowing about it, the CLOSE_WAIT status message appears on the requester's side, and the FIN_WAIT_2 message appears on the broker's or enterprise server's side.

You can use the CloseWaitTimeout in the mgreq.ini file to control this behavior.

3.4 Timeouts

uniPaaS provides various timeouts defined in the INI files, which can be helpful when troubleshooting. For more information about the timeouts, see the *uniPaaS Help*.

MGREQ.INI / Module using it:	REQUESTER	BROKER	APP SERVER
BrokerTimeout	√		
RequesterTimeout	√		
CommTimeout	√	√	√
ServerTimeout		√	
CloseWaitTimeout	√		
ShutdownTimeout		√	

3.5 Log File Settings

There are three logs:

- **Mgreq.ini** - Displays low-level activities, such as TCP/IP, threads, events, etc. For more information about the Mgreq.ini file, see the *uniPaaS Help*.

- **Mgrb.ini** - Displays high-level broker activities such as initialization, receiving requests, locating enterprise servers, assigning enterprise servers to requesters, and so on. This log helps you understand whether or not the broker accepted a request, and what was done with the request. For more information about the Mgrb.ini file, see the *uniPaaS Help*.

- **Mrb_event.log**, which is issued from the broker and is not related to any INI file, registers significant broker activities, such as startup and shutdown of the broker and enterprise servers, etc.

Appendix I - uniPaaS's Information and Error Codes

Information

Error Number	Mnemonic	Troubleshooting
0	RQ_OK	
-1	RQ_INF_TERMINATE	
-2	RQ_INF_TERMINATE_THREAD	The thread itself is terminated.
-3	RQ_INF_RECONNECT_MAIN	
-4	RQ_INF_RETRY	Internal status code. For a requester – when the broker instructs the requester to retry. For an engine – when the engine retries a connection to the broker. For the broker – when retrying a submission of an asynchronous request. In all cases, unless this status code is followed by another error status code, it should be ignored.
-5	RQ_INF_LOG_ACTIVE	
-6	RQ_INF_ALREADY_INITIALIZED	
-10	RQGNRC_INF_NOWAIT	
-11	RQGNRC_INF_NO_RESULT	No output returned from remote program. This status is returned if a program executed (in the Runtime engine) by a request did not return any content to the requester. A program can return content in one of several ways: (1) Form Output to an I/O of type 'Requester'. (2) Calling the File2Req or Blob2Req functions.
-32	RQMRG_INF_NO_REQUEST	Internal status codes of the enterprise server.
-33	RQMRG_INF_IN_PROGRESS	
-34	RQMRG_INF_CLOSE_APPSERV	
-35	RQMRG_INF_WARNING_ERRS_ON_INIT	
-40	RQMRB_INF_NO_PND_REQ	Internal status codes of the broker.
-41	RQMRB_INF_APP_NOT_FOUND	
-42	RQMRB_INF_APP_IN_USE	
-43	RQMRB_INF_RT_NOT_TERM	
-44	RQMRB_INF_ACK_SENT	

-45	RQMRB_INF_CNCT_MAIN_REFUSED	
-46	RQMRB_INF_CNCT_MAIN_NOT_RSPND	
-50	MM_INF_LAST_BIGGER	Low-level status code of the memory tables, which are the foundation for broker resource management.
-51	MM_INF_LAST_SMALLER	
-52	MM_INF_NO_REC	
-53	MM_INF_EOF	
-54	MM_INF_FILTER_LIMIT	
-60	RQTCP_INF_TIMEOUT	Can occur in several low-level scenarios. In each scenario, it will be handled differently. For example: i. A requester submits a request to the broker for an unsupported application, which does not exist, error –103; or is busy, error -104. The broker issues an acknowledgement message, and the requester continues to wait for an available enterprise server, according to the BrokerTimeout value in the Mgreq.ini file or its equivalent in the Servers table of the Magic.ini file. If no enterprise server becomes available in the specified period of time, the requester receives status -60 from the TCP/IP layer, which is then handled as status code -103 or -104. ii. A requester submits a request to the broker and does not receive any response from the broker, which is handled as error -105. iii. A requester contacts an enterprise server, submits a request to the enterprise server, and begins to wait for the response. If a request timeout was set and the enterprise server did not complete the request, the requester receives status -60 from the TCP/IP layer, which is then handled as status -110 (REQUEST-TIMEOUT).

Errors

-102	RQGNRC_ERR_CNCT_REFUSED_MRB	The connection to the broker was refused. Verify that the broker was started on the port on the specified machine. Refer to its Mgrb.ini file.
-103	RQGNRC_ERR_APP_NOT_FOUND	The engine does not support the application; or the Appl=entry in the Mgreq.ini file was specified and the required application does not appear in the list of applications registered to the selected broker. Use the Broker monitor to view the status of enterprise servers.
-104	RQGNRC_ERR_APP_IN_USE	All enterprise servers supporting the application are busy serving other requests. This error message can also appear if the Broker Filter limit is exceeded (and the enterprise server is not completely busy). These status codes as well as the previous one are controlled by a broker timeout keyword. For the uniPaaS client, the timeout value is in the Servers table. In the Mgreq.ini file, the timeout value is the BrokerTimeout parameter. This error usually appears after 10 seconds, the default setting, which is the timeout set by the broker to deal with a synchronous request. To solve this time-out: 1. Edit the Mgreq.ini file in the Scripts directory for Internet requests, or the Magic.ini file for running Call Remote operations. Set the BrokerTimeout value in the Mgreq.ini file to a value higher than 10 seconds, for example 300 seconds (5 minutes). 2. Alternatively, in the broker's Magic directory, increase the number of Magic engines to start in the MRB_EXECUTABLES_LIST section of the Mgrb.ini file. The engines may be deployed in the background. You must restart the broker by modifying the Mgrb.ini file. 3. If your enterprise server engines, Web server, and broker are loaded on different machines, you must also make sure that all machines involved in supporting the Web application can communicate with one another using a host name and not only IP addresses. You can test this using the ping command. If required, you should modify the host file on each machine so that they can communicate with each other using host names. (This configuration is not highly recommended.) 4. Use the Broker monitor to view the status of the enterprise servers. When dealing with a J2EE environment, the EJB continues to connect to the enterprise server for as long as allowed in the CommunicationTimeout setting. If the EJB fails to connect, it will throw the Application Busy exception to the client that activated it.
-105	RQGNRC_ERR_MRB_NOT_RSPND	The broker did not return any response to a requester, not even an ACK message. Refer to test case -105: Broker not responding. For query and administrative requests, you can increase the Broker timeout value.
-106	RQGNRC_ERR_RT_NOT_RSPND	A requester was not able to send a request to an engine that was assigned by the broker.

		Verify that the engine is alive (in the Task Manager, according to the PID displayed in the broker's status window); Call this engine from a command line requester to isolate the source of the problem. (Connections are created and destroyed for each request of the command line requester.)
-107	RQGNRC_ERR_CNCT_RESET	Connection reset by the enterprise server. This message appears when: ○ The enterprise server was aborted abnormally during the execution of a request. ○ The connection was reset due to network connection problems. Refer to test case -105: Broker not responding, and to status code -144 below. Verify that the engine is alive; (for example, in the Task Manager, according to the PID displayed in the broker's status window). Call this engine from a command line requester to isolate the source of the problem. (Connections are created and destroyed for each request of the command line requester.)
-108	RQGNRC_ERR_INVALID_REQ_HDL	When the requester API is used, it may be related to a wrong argument that was passed to the API.
-109	RQGNRC_ERR_CNCT_REFUSED_RT	Communication problems between requesters and enterprise servers. Check the firewall settings between client and server computers, host name resolution, etc. You can view the Broker monitor to view the status of the enterprise servers. Ping the host name and IP address of the assigned enterprise server. Verify that the engine is alive; for example, in the Task Manager, according to the PID displayed in the broker's status window. Call this engine from a command line requester, in order to isolate the source of the problem (Connections are created and destroyed for each request of the command line requester.)
-110	RQGNRC_ERR_REQUEST_TIMEOUT	The execution of the task was not completed during the Request Timeout interval. Increase the Requester Timeout keyword (in the Mgreq.ini file for Internet or command line, or in the Magic.ini file for Call Remote).
-111	RQGNRC_ERR_NOT_MRB	A requester tried to connect to a TCP/IP server that is not a Magic Broker.
-112	RQGNRC_WRN_ALT_MRB	Obsolete status code.
-113	RQGNRC_ERR_APPNAME_REQUIRED	An application name was not specified in a Call Remote operation.
-114	RQGNRC_ERR_PRGNAME_REQUIRED	A program name was not specified in a Call Remote operation.
-117	RQGNRC_ERR_RMC_DISABLED_FOR_J2EE	J2EE type servers can only accept remote calls from EJBs. This error occurs following a remote call by a different requester. Set Gateway=5 in the Mgreq.ini file to ensure that users cannot send remote calls from a command line or Web requester in this directory.

-118	RQGNRC_ERR_TIME_STAMP	A requester received a response designated for another request from the broker. This is a severe error that should never be found in log files. As an interim solution, increase the Broker Timeout's value.
-128	RQMRI_ERR_APP_REJECTED	Two or more requests attempted to open different applications in the same engine when either no application was open or no context existed in that engine. Retry the request. When dealing with a J2EE environment, the EJB continues to connect to the enterprise server for as long as specified in the CommunicationTimeout setting. If the EJB fails to connect, it throws the “ApplicationBusy” exception to the client.
-130	RQMRI_ERR_APP_OPEN_FAIL	The server engine could not open the application. Check that the application can be opened locally (for example, online).
-131	RQMRI_ERR_BAD_PRG	The enterprise server could not find the requested program. Check the program's public name.
-133	RQMRI_ERR_ACCESS_DENIED	Access is denied to the application. This error can occur when: • A wrong user or password was passed to the enterprise server. • The user had no rights to execute the program.
-134	RQMRI_ERR_LIMITED_LICENSE_PART	License limited to only partitioning requests.
-135	RQMRI_ERR_LIMITED_LICENSE_HTTP	License limited to only Internet requests.
-136	RQMRI_ERR_LIMITED_LICENSE_CS	Maximum number of hits was reached during license validation. When using a non-server license, the enterprise server is limited to 2,000 requests. Use a server license.
-137	RQMRI_ERR_REQ_REJECTED	The enterprise server cannot execute the request because of a timing problem. This error is usually related to switching between Runtime and Toolkit modes.
-138	RQMRI_ERR_RT_ERROR_MSG	During the execution of a program in the enterprise server, the program did not complete properly; for example, a verify error that aborted execution or any other abort condition. If the executed program failed to complete, these error messages were trapped by the enterprise server and sent back to the requester. If the executed program was executed successfully despite the error messages, the program’s output is returned, overriding any error messages. When the requester is an Internet requester, the error messages are sent to the remote browser. When the requester is a command line requester, the error messages are displayed in the console. When the requester is an eDeveloper 9 engine, this error message is not displayed. Check the executed programs and its descendants by pressing F8. When dealing with a J2EE environment, the EJB includes error messages from the aborted program in an exception thrown to the EJB client.
-139	RQMRI_ERR_THREAD_ABORTED	During the execution of a program, the program terminated abnormally. In the MAGIC_SPECIALS section, set the

		ExceptionMessageBoxDisplay flag to Yes, and use debugging techniques, such as WinDbg, to find the problem.
-140	RQTCP_ERR_NOT_INITIALIZED	Refer to winsock error 10093 below.
-142	RQTCP_ERR_BIND_FAIL	A server module (e.g. broker or enterprise server) failed to bind to a local address, which might already be used.
-143	RQTCP_ERR_CNCT_REFUSED	A connection from a client module to a peer was refused. This can happen between almost any two modules. For example, from a requester to the broker or enterprise server, or from an enterprise server to the broker.
-144	RQTCP_ERR_CNCT_RESET	An established connection was reset. The connection is no longer valid and can no longer be used. Refer to winsock error 10054 and to test case #1 for typical scenarios.
-146	RQTCP_ERR_BIND_HOST_NOT_FOUND	A server module cannot bind to a local address due to an unresolved name; for example, as specified by /LocalHost in the Magic.ini/communications/tcpip The Local Host entry in the Mgreql.ini file or /LocalHost in the TCP/IP parameters in the Magic.ini file (for example, TCP/IP = 2,30,1500-2000 /LocalHost=myserver) specifies an invalid host name.
-147	RQTCP_ERR_CNCT_HOST_NOT_FOUND	Unknown host. A client module cannot establish a connection to a server module due to an unresolved name specified for the server. This can occur when a requester on one computer receives an enterprise server address from the broker, and the enterprise server is located on a host whose name is known to the broker but not to the requester. A solid DNS/DHCP configuration usually prevents this scenario. This error is similar to ERR-BIND_HOST_NOT_FOUND. A requester cannot connect to an unknown broker or enterprise server. You should check the MessagingServer keyword in the Mgreq.ini file. The broker address must contain the Internet address, such as 88.0.184/2001.
-148	RQTCP_ERR_CNCT_CLOSED	A connection was unexpectedly closed by a peer. Refer to the relevant appendix (at the end of the document).
-149	RQTCP_ERR_OUT_OF_SOCKETS	The current module reached the maximum number of opened sockets (the default is 1000). Increase this value using the keyword Handles=NNNN in the Mgreq.ini file.
-150	RQMRG_ERR_CNCT_REFUSED_MRB	An enterprise server could not connect to the broker. Check if the broker has started and that the host name of the broker MessagingServer keyword in the Magic.ini file belongs to the correct IP address (ping <mrbhost>). You can trace the problem by setting the Log parameter to Enabled in the Mgreq.ini file (for example, log = reg.log Y R) in the enterprise server directory.
-151	RQMRG_ERR_CNCT_CLOSED_BY_REQ	During the execution of a request, the requester closed the connection after received status -110 (REQUESTER_TIMEOUT). As a result, no output was sent back from the enterprise server to the requester. Check the requester in the client machine and the enterprise server. If possible, reproduce the problem with log “enabled” in the Mgreq.ini file (log = req.log Y R) in the directories of the requester and the enterprise

		server. This is an internal status code that closely coupled with status -110 and should be handled on the client side (refer to status –110).
-156	RQMRG_ERR_OUT_OF_SEQ_MSG	The session between the client and the server is no longer synchronized. A context serving a client received an event from the client containing an unexpected session counter. (Each request from a client must have a session counter equal to the previous session counter + 1.) If possible, reproduce the problem by setting the Log parameter to Enabled in the Mgreq.ini file (log = req.log Y R) in the directories of the requester and the enterprise server.
-157	RQMRG_ERR_AUTHENTICATION	The credentials sent from the RIA client could not be authenticated by the server, because of an unknown user, wrong password, etc.
-160	RQSPAWN_ERR_EXE_NOT_FOUND	Status codes related to spawning executable files, usually by the broker. The executable file name may be wrong, the file may be damaged, the username or password may be wrong, etc.
-161	RQSPAWN_ERR_PATH_NOT_FOUND	
-162	RQSPAWN_ERR_BAD_EXE	
-163	RQSPAWN_ERR_BAD_LOGIN	
-164	RQSPAWN_ERR_PRIVILEGE_NOT_HELD	
-165	RQSPAWN_ERR_ARG_BIG	
-166	RQSPAWN_ERR_MODE_EINVAL	
-167	RQSPAWN_ERR_NOMEM	
-168	RQSPAWN_ERR_NOPROCESS	
-169	RQSPAWN_ERR_NET_UNREACHABLE	
-170	MM_ERR_INV_SEG	Memory table status codes, relevant only during the broker's processing. (The memory tables were originally developed to manage the broker resources.)
-171	MM_ERR_DUPLICATE	
-172	MM_ERR_INV_OPER	
-173	MM_ERR_INV_POS	
-174	MM_ERR_NO_INIT	
-175	MM_ERR_TARGET_EXISTS	
-176	MM_ERR_OUT_OF_HDLS	
-177	MM_ERR_KEY_DISABLED	
-180	RQMRB_WRN_EXE_NOT_FOUND	The broker was requested to spawn an executable that is not listed in its [APPLICATIONS_LIST].
-181	RQMRB_WRN_RT_NOT_FOUND	The broker was requested to perform an operation on an engine that was not registered (host/port), for example to terminate an engine. If possible, reproduce the problem with log "enabled" in the Mgrb.ini file (log = mrb.log Y R).
-182	RQMRB_WRN_REQ_NOT_FOUND	The broker was requested to perform an operation on an unknown request, for example to modify its priority. If possible, reproduce the problem by setting the Log parameter to Enabled in the Mgrb.ini file.
-183	RQMRB_WRN_REQ_NOT_MATCH	The broker was requested to perform an operation on a request that does not match the requester application name.

		If possible, reproduce the problem by setting the Log parameter to Enabled in the Mgrb.ini file.
-184	RQMRB_ERR_INI_NOT_PROTECTED	Obsolete status code
-185	RQMRB_ERR_REGISTER_SERVICE	The broker failed during its initialization as a service. Reproduce the problem by setting the Log parameter to Enabled in the Mgrb.ini file.
-186	RQMRB_ERR_REPORT_SERVICE_STATUS	The broker failed during its initialization as a service. Reproduce the problem by setting the Log parameter to Enabled in the Mgrb.ini file.
-187	RQMRB_ERR_CNCT_REFUSED_REMOTE_MRB	The broker failed to connect a remote broker in order to start a remote executable ([MRB_REMOTE_EXECUTABLES_LIST]). Verify that the address of the remote broker is valid (host/port), and that a remote broker is bound to that address.
-197	RQMRB_ERR_CTX_NOT_FOUND	Context not found. Refer to test case -197: Context Not Found.
-198	RQMRB_ERR_QUE_LIMIT	Queue limit reached. Increase the QueueMaxSize in the Mgrb.ini file.
-200	RQ_ERR_UNEXPECTED	Unexpected error. When accompanied by other error codes, this error must be addressed by R&D.
-201	RQ_ERR_NOT_INITIALIZED	Code partitioning error. TCP/IP services were not installed. Install TCP/IP.
-202	RQFIO_ERR_OPEN_RESULT_FILE	1. The requester asked that the output be written into a file (either using a field in the Call Remote dialog box in an eDeveloper 9 client, or using keywords in the Mgreq.ini file or in Mgrqcmdl –file=). 2. The combined file name (directory and/or file name) is illegal.
-203	RQLIB_ERR_INI_FILE	An INI file could not be opened.
-204	RQCMDL_ERR_BAD_ARGS	The command line requester could not parse its arguments. Use the help notes (displayed when activated without any argument).
-205	RQ_ERR_WRONG_MSG_SRVR	A query was requested on middleware that does not support queries; for example when starting the engine to server requests from EJB.
-206	ERR_SOAP_SRVER_PARSE	An incoming SOAP envelope contained invalid element(s). You can use an HTTP tracer to locate and fix the invalid element(s).
-210	RQMRILOW_ERR_RECV_FAIL	An internal error was caught while receiving a message from a peer. If possible, reproduce the problem by setting the Log parameter to Enabled in the Mgreq.ini file (log = req.log Y R).
-211	RQMRILOW_ERR_NOT_MRI	An internal error was caught while receiving a message from a peer. **Scenario #1:** The peer was not a uniPaaS module; a message was sent by another TCP/IP module, such as Internet Explorer. **Scenario #2:** The peer did not set the EncryptedCommunication keyword in the Mgreq.ini file or set a mismatched value in the keyword.

-212	RQMRILOW_ERR_OLD_MRI	An internal error was caught while receiving a message from a peer. The peer was an out-of-date module (e.g. a Magic8 engine replying to a Magic9 requester). If possible, reproduce the problem by setting the Log parameter to Enabled in the Mgreq.ini file (log = req.log Y R). Verify that the replying module is a module of the same version as the local module.
-260	RQHTTP_ERR_UPLOAD_TOO_BIG	The size of an uploaded file from a browser to the requester exceeded the maximal size. Increase MaxUploadKB in the Mgreq.ini file in the scripts directory.

Appendix II - DB Errors

When you run the "**Mgrqcmdl –query=log**" command, DB error, warning, or informational codes may be returned:

Error Number	Mnemonic	Meaning
1	DB_ERR_REC_LOCKED	Record is locked.
2	DB_ERR_DUP_KEY	Duplicate key.
3	DB_ERR_CONSTR_FAIL	Constraint failure.
4	DB_ERR_TRIGGER_FAIL	Trigger failure.
5	DB_ERR_REC_UPDATED	Record was updated.
6	DB_ERR_NO_ROWS_AFFECTED	Record was updated by another user.
7	DB_ERR_UPDATE_FAIL	Record update failed.
9	DB_ERR_EXEC_SQL	Error executing the SQL command.
10	DB_ERR_BAD_SQL_CMD	Invalid SQL command.
11	DB_ERR_BADINI	Database initialization failed.
12	DB_ERR_BADNAME	Invalid table name.
13	DB_ERR_DAMAGED	Damaged table.
15	DB_ERR_BADOPEN	Table could not be opened.
16	DB_ERR_BADCLOSE	Failed to close table.
17	DB_ERR_RSRC_LOCKED	Waiting for lock in database.
18	DB_ERR_REC_LOCKED_NOBUF	Waiting to fetch locked row.
19	DB_ERR_NODEF	Database definition could not be loaded.
20	DB_ERR_REC_LOCKED_NOW	Record is locked.
23	DB_ERR_READONLY	Table opened for read and attempt was made to write to it.
25	DB_ERR_CAPACITY	Valid only when using Demo license.
26	DB_ERR_TRANS_COMMIT	Commit transaction operation failed.
27	DB_ERR_TRANS_OPEN	Begin transaction failed.
28	DB_ERR_TRANS_ABORT	Rollback transaction operation failed.
29	DB_ERR_BADDEF	Definition mismatch.
30	DB_ERR_INVALID_OWNR	Invalid access key to table.
31	DB_ERR_CLR_OWNR_FAIL	Failed to remove access key.

32	DB_ERR_ALTER_TBL	Database failed to alter table.
33	DB_ERR_SORT_TBL	Database failed to sort table.
34	DB_ERR_CANOT_REMOVE	Table could not be deleted.
35	DB_ERR_CANOT_RENAME	Table cannot be renamed.
37	DB_ERR_TARGET_FILE_EXIST	Table creation failed. Table already exists.
38	DB_ERR_FILE_IS_VIEW	Table is a view.
39	DB_ERR_CANOT_COPY	Cannot create; drop, or copy a view.
40	DB_ERR_STOP	Error executing the SQL command.
41	DB_ERR_STR_BAD_NAME	Invalid table name.
43	DB_ERR_BAD_QRY	iSeries – Invalid Open Query.
46	DB_WRN_CACHE_TOO_BIG	Not enough memory. Table cache was not started.
47	DB_ERR_LOSTREC	Record was lost.
48	DB_ERR_FILE_LOCKED	Unable to lock table.
49	DB_ERR_MAX_CONN_EX	Maximum connections reached.
50	DB_ERR_DEADLOCK	Deadlock.
51	DB_ERR_BADCREATE	Create error.
52	DB_ERR_FIL_NOT_EXIST	Table does not exist.
54	DB_ERR_IDX_CREATE_FAIL	Table index cannot be created.
55	DB_ERR_CONNECT_FAIL	Cannot connect to the database.
56	DB_ERR_FATAL	Unknown fatal error.
57	DB_ERR_INSERT_FAIL	Record cannot be inserted.
58	DB_ERR_DELETE_FAIL	Record cannot be deleted.
60	DB_ERR_NOREC	No more records in table (usually returned when reaching the end or start of table, depending on the order of the retrieval).
61	DB_ERR_NOT_EXIST	Table does not exist.
62	DB_ERR_GET_USR_PWD	Incorrect database password.
64	DB_ERR_NOTSUPPORT_FUNC	iSeries – unsupported function in the **Magic Where** expression.

Appendix III - Winsock Errors

The following list of errors should usually be reported to Technical Support. An exception to this is error 10054, connection reset by peer, which can occur between the web requester and the browser if the browser was closed during the execution of a request. This case cannot be regarded as an error and can be ignored.

Error Number	Mnemonic	Meaning
0	WSABASEERR	No error.
10004	WSAEINTR	Interrupted system call.
10009	WSAEBADF	Bad file number.
10013	WSAEACCES	Permission denied.
10014	WSAEFAULT	Bad address.
10022	WSAEINVAL	Invalid argument.
10024	WSAEMFILE	Too many open files.
10035	WSAEWOULDBLOCK	Operation would block.
10036	WSAEINPROGRESS	Operation now in progress.
10037	WSAEALREADY	Operation already in progress.
10038	WSAENOTSOCK	Socket operation on non-socket.
10039	WSAEDESTADDRREQ	Destination address required.
10040	WSAEMSGSIZE	Message too long.
10041	WSAEPROTOTYPE	Protocol is wrong type for socket.
10042	WSAENOPROTOOPT	Bad protocol option.
10043	WSAEPROTONOSUPPORT	Protocol not supported.
10044	WSAESOCKTNOSUPPORT	Socket type not supported.
10045	WSAEOPNOTSUPP	Operation not supported on socket.
10046	WSAEPFNOSUPPORT	Protocol family not supported.
10047	WSAEAFNOSUPPORT	Address family not supported by protocol family.
10048	WSAEADDRINUSE	Address already in use.

10049	WSAEADDRNOTAVAIL	Cannot assign requested address.
10050	WSAENETDOWN	Network is down.
10051	WSAENETUNREACH	Network not accessible.
10052	WSAENETRESET	Net dropped connection or reset.
10053	WSAECONNABORTED	Software caused connection abort.
10054	WSAECONNRESET	Connection reset by peer.
10055	WSAENOBUFS	No buffer space available.
10056	WSAEISCONN	Socket is already connected.
10057	WSAENOTCONN	Socket is not connected.
10058	WSAESHUTDOWN	Cannot send after socket shutdown.
10059	WSAETOOMANYREFS	Too many references; cannot splice.
10060	WSAETIMEDOUT	Connection timed out.
10061	WSAECONNREFUSED	Connection was refused.
10062	WSAELOOP	Too many levels of symbolic links.
10063	WSAENAMETOOLONG	File name too long.
10064	WSAEHOSTDOWN	Host is down.
10065	WSAEHOSTUNREACH	No route to host.
10066	WSAENOTEMPTY	Directory not empty.
10067	WSAEPROCLIM	Too many processes.
10068	WSAEUSERS	Too many users.
10069	WSAEDQUOT	Disk quota exceeded.
10070	WSAESTALE	Stale NFS file handle.
10091	WSASYSNOTREADY	Network subsystem not available.
10092	WSAVERNOTSUPPORTED	WINSOCK DLL version out of range.
10093	WSANOTINITIALISED	Successful WSASTARTUP not yet performed.
10071	WSAEREMOTE	Too many remote levels in path.

11001	WSAHOST_NOT_FOUND	Host not found.
11002	WSATRY_AGAIN	Non-authoritative host not found.
11003	WSANO_RECOVERY	Non-recoverable errors: FORMERR, REFUSED NOTIMP
11004 *	WSANO_DATA	Valid name, but no data record of requested type.
11004 *	WSANO_ADDRESS	No address. Look for MX record.

Appendix IV - Test Cases

-105: Broker not responding

The broker should immediately respond to any request from a requester, even if no enterprise server is available to serve the application required by the requester. In this case, the broker must respond with an acknowledgement message (ACK), and the requester must continue to wait, according to the broker timeout value.

If the broker fails to respond on time, the requester receives a status code (-105) message from the generic messaging layer (Mgrqgnrc.dll).

Procedure

If the broker is heavily loaded in terms of CPU or memory, the first action item is to increase the CommTimeout keyword in the Mgreq.ini file for both the requester and the broker.

Another action item is to relocate the broker to another stronger, and preferably dedicated, host.

Another action item is to locate the broker on the same host as the requester, and to have the enterprise servers, which usually take most of the load, on other hosts.

-138: Runtime crash

During execution of a program on the enterprise server, the program did not complete properly; for example, there was a verify error that aborted execution, or another abort condition.

The enterprise server collects the error messages during the program's execution and sends them back to the requester. The web requesters then display these messages as an HTML error page for the clients' information.

Note: When a program completes successfully, nothing is sent back to the requester and error messages are discarded.

Procedure

Check the executed programs and its descendants by pressing F8. This should only be done on the client.

-144: Low-level connection reset

If the connection from a requester or enterprise server to the broker was reset, the **mrb_event.log** displays an error in the following format:

1092 23:29:54,03534 01/05/2002 Error: "TCP/IP error: Connection reset" (-144) (server2/1501)

1092 - Thread id (internal to the broker). This is used as a starting point for debugging to synchronize with other logs, such as those from the Mgrb.ini or Mgreq.ini files.

(server2/1501) - The module with which the connection was reset. If the problematic module is an enterprise server, this is the enterprise server address, as registered during the enterprise server startup. When the problematic module is a requester, such as when the computer hosting IIS crashed for some reason, no address is displayed.

Procedure

If the problematic module is an enterprise server:

- The first action item is to view the Task Manager or its equivalent in other operating systems.
- If the enterprise server was removed from the tasks list, the problem should be further researched as an enterprise server problem, using logs. The broker's history log, either **Mgrqcmdl -query=log** or **the log from the broker monitor** application, can help focus on the program that was activated when the enterprise server crashed.
- If the enterprise server was not removed from the tasks list, the problem is likely to be a partitioning / TCP/IP failure where the broker mistakenly received a connection that was reset from the enterprise server. In this case, a log file in the Mgreq.ini file can provide a starting point for debugging from both sides, the broker's and the enterprise server's.

Note: In addition to the **mrb_event.log**, the requester receives error -107 when its enterprise server crashed during request execution.

-197: Context Not Found

A context-id sent by a browser client does not exist in the Enterprise Server.

Whenever a browser client session starts, the Enterprise Server generates a unique context-id for that session. This context-id links subsequent requests from the browser client to the Enterprise Server.

The Enterprise Server keeps the context alive according to the value specified in the **ContextInactivityTimeout** environment parameter in the Magic.ini file. This timeout value measures the time since the last request and times out the request if it is greater than this value.

Another environment parameter that may influence the session between the browser client and the Enterprise Server is **ContextUnloadTimeout.**

Procedure

Scenario 1

During a Browser Client session, the end-user had no interaction with the Enterprise Server for a period longer than the value set by the ContextInactivityTimeout environment setting.

Solution: Carefully increase this environment value. Setting a larger value means that the Enterprise Server maintains more open contexts for a longer time, which has a negative influence on available resources. Remember that the ContextInactivityTimeout environment setting is specified in tenths of seconds, and the default is 36000 or 1 hour.

Scenario 2:

A URL containing a context-id was accessed after the context expired.

Solution: The URL should access a program that starts a browser-based session; for example, **appname=myapp&prgname=myprg**.

Scenario 3

While running a browser client session, the end-user selected another URL and then, using the Internet Explorer's Back functionality, tried to return to the uniPaaS browser session after the **ContextUnloadTimeout** had expired.

Solution: Carefully increase this environment value. Setting a larger value means that the Enterprise Server maintains more open contexts for a longer time, which has a negative influence on available resources.

Remember that the ContextUnloadTimeout environment setting is specified in tenths of seconds, and the default is 1200 or 2 minutes.

Remember that in development mode this timeout is limited to 1/10 of a second. Whenever a Browser program has been started using F7 unloads, meaning that the browser is closed or switched to another URL, the developer is toggled back to Toolkit again.